

云南大学数学系《运筹学通论实验》课程上机实验报告

课程名称：运筹学通论	学期：2015-2016 学年第二学期	成绩：
指导教师：李建平	学生姓名：金洋	学生学号：20131910023
实验名称： Simplex Method & Two-phase Method		
实验编号：No.2	实验日期： 2016/3/25	实验学时： 1
学院： 数学与统计学院	专业： 信息与计算科学	年级： 2013

一、实验目的

使用 c 语言实现单纯形法和两阶段法求解线性规划问题；

二、实验内容

1. 例 2-6 用单纯形法求解线性规划问题

$$\begin{aligned} \max z &= 2x_1 + 3x_2 + 0x_3 + 0x_4 + 0x_5 \\ \begin{cases} x_1 + 2x_2 + x_3 & = 8 \\ 4x_1 & + x_4 = 16 \\ 4x_2 & + x_5 = 12 \\ x_j \geq 0, j = 1, 2, \dots, 5 \end{cases} \end{aligned}$$

2. 例 2-9 线性规划问题

$$\begin{aligned} \min z &= -3x_1 + x_2 + x_3 \\ \begin{cases} x_1 - 2x_2 + x_3 \leq 11 \\ -4x_1 + x_2 + 2x_3 \geq 3 \\ -2x_1 & + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

试用两阶段法求解；

3. 例 2-10（合理利用线材问题）

$$\begin{aligned} \min z &= 0x_1 + 0.1x_2 + 0.2x_3 + 0.3x_4 + 0.8x_5 \\ \begin{cases} x_1 + 2x_2 & + x_4 = 100 \\ & 2x_3 + 2x_4 + x_5 = 100 \\ 3x_1 + x_2 + 2x_3 & + 3x_5 = 100 \\ x_1, x_2, x_3, x_4, x_5 \geq 0, \text{且为整数} \end{cases} \end{aligned}$$

试用两阶段法求解，当目标函数改为等价的 $z = x_1 + x_2 + x_3 + x_4 + x_5$ 时,解是否

一致;

4. 作业题 2.6 (2), 用两阶段法求解下述 LP 问题:

$$\begin{aligned} \min z &= 2x_1 + 3x_2 + x_3 \\ \begin{cases} x_1 + 4x_2 + 2x_3 \geq 8 \\ 3x_1 + 2x_2 \geq 6 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

5. 例 3-4 求解线性规划问题

$$\begin{aligned} \max z &= x_1 + x_2 \\ \begin{cases} -x_1 + x_2 + x_3 \leq 2 \\ -2x_1 + x_2 - x_3 \leq 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

三、使用环境

平台: Microsoft Visual C++ 6.0

语言: C 语言

四、算法介绍

1. 单纯形法

Algorithm Simplex

Input 标准型下的相关信息——

N: the number of decision variables;

M: the number of constraint conditions;

$C_{1 \times N}$: value coefficient of the objective function;

$A_{M \times N}$, $b_{M \times 1}$: the constraint conditions ;

$BASIS_{M \times 1}$: 初始基可行解基变量所在的列号;

Output

原 LP 的标准型;

求解结果 $X[i](i=1,2,\dots,N)$;

解的类型;

Begin

Step 0: Input and initialize;

记录 x_j 是否为基;

Output 原 LP 的标准型;

Step 1: 计算各列的检验数 $\sigma[j]$, $j=1, \dots, N$

Step 2: For $i=1$ through N

 If $\sigma[i] > 0$ then break;

$k=i$; (x_k 为换入变量)

Step 3: If (All $\sigma[i] \leq 0$) then

 If (基变量中有人工变量) then

 1. Output “无可行解”;

 2. Return 0;

 Else

 1. Output “最优解”;

 2. Output 当前已求得的最优解;

 3. Return 1;

Step 4: if ($\sigma[j] > 0$) and ($A_j \leq 0$) (for some j)

 1. Output “无界解”;

 2. Return 0;

Step 5: $l = 0$, $\theta = MAXNUM$;

Step 6: For $i=1$ through M

 If ($A[i][k] > 0$) then

 1. $t = b[i]/A[i][k]$;

2. If ($t < \theta$) then

1. $\theta = t$;

2. $l = i$

即在单纯形表中 l 行对应的基变量 为换出变量;

$A[l][k]$ 为主元素;

Step 7: 修改基变量;

Step 8: 旋转变换;

Step 9: repeat Steps 1~8 ;

End.

2. 两阶段法

前一部分着重描述了单纯形法的算法, 从输入上对比, 单纯形法对输入数据的要求较高——需要输入标准型, 且已得到一组基可行解, 这对于一般 LP 问题, 需要事先进行人工变形; 两阶段法可以完美解决以上问题, 其核心算法是两次运用单纯形法, 以下对其的算法介绍, 着重描述了两阶段法的初始化和两次调用单纯形法, 而不再重复写单纯形法的详细步骤。

Algorithm Two-phase Method

Input 原 LP 的相关信息——

n : the number of decision variables;

M : the number of constraint conditions;

DIRE[0]: 约束条件中 " \leq " 类条件数;

DIRE[1]: 约束条件中 " \geq " 类条件数;

DIRE[2]: 约束条件中 " $=$ " 类条件数;

objType: 目标函数类型(1 for max、0 for min);

$C_{1 \times N}$: value coefficient of the objective function;

$A_{M \times N}$, $b_{M \times 1}$ the constraint conditions (在输入约束条件时, 按照 " \leq "

类、" \geq "、" $=$ " 类的顺序输入);

Output

原 LP 的标准型;

加入人工变量后, 第一阶段标准型;

第一阶段解的类型、求解结果 $X[i](i=1,2,\dots,N_A)$; (N_A 为原 LP 决策变量个数 n 、剩余变量、人工变量之和);

第二阶段解的类型、求解结果 $X[i](i=1,2,\dots,N)$; (N 为原 LP 决策变量个数、剩余变量之和);

Begin

Step 1: $N=n+\text{DIRE}[0]+\text{DIRE}[1]$; //原 LP 标准型下决策变量总个数

$NA=N+\text{DIRE}[1]+\text{DIRE}[2]$; //加入人工变量后决策变量总个数

If ($N==NA$) then

1. Output “输入形式的 LP 经变形未产生人工变量,请变形后
再用两阶段法求解”;

2. Exit(0) 结束程序

Step 2: If (!objType) then $C=-C$; //原 LP 求解 min 时,需转化为求解 max,
这样可直接调用单纯形法 Simplex()

Step 3: 构造仅含人工变量的目标函数(价值系数矩阵为 $C1$);

加入人工变量, 形成初始基可行解;

Step 4: Output 原 LP 的标准型;

Output 加入人工变量后, 第一阶段标准型;

Step 5: firstPhase=Simplex($C1,N,NA$); //第一阶段, 返回值判断第一阶段
是否存在最优解;

Step 6: if (firstPhase) then

Simplex(C,N,N); //第二阶段

else

Output "原 LP 无可行解";

End.

五、调试过程

1. 程序代码

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define LENF sizeof(float)
#define LENINT sizeof(int)
#define MAX 1000000

int N;      //N 为决策变量个数
int NA;     //NA 为加上人工变量后决策变量总个数
int M;      //M 为约束条件数（不含非负条件），默认所有决策变量为非负
int objType;//目标函数类型 1 for Max;0 for Min;
float *C;   //价值系数
float *C1;  //两阶段法第一阶段的价值系数
float **A;  //约束条件系数矩阵
float *b;   //约束条件限额系数
int *BASIS; //储存基所在的列数
int *STATUS;//基变量状态,提示是否是基变量
float *SIGMA; //检验数 $\sigma$ 
int DIRE[3]; //分别代表约束条件中 $\leq$ , $\geq$ , $=$ 的条件个数

/*单纯形法初始化*/

void initSimplex() {
```

```
int i,j;

printf("\n\n");

printf("请输入线性规划问题相关信息(已化为标准型,且已知初始基可行解基
变量所在的列号):\n");


printf("请输入决策变量个数 N=");

scanf("%d",&N);

printf("请输入约束条件(已默认所有决策变量为非负)个数 M=");

scanf("%d",&M);


/*动态分配了数组 C[N],目标函数价值系数*/
C=(float*) malloc((N+1)*LENF);

printf("请输入目标函数价值系数:\n");

for (i=1;i<=N;i++)

    scanf("%f",&C[i]);


/*动态分配了数组 A[M][N],约束条件系数*/
A=(float**)malloc((M+1)*sizeof(float*)); //第一维

for(i=1;i<=M; i++)

    A[i]=(float*)malloc((N+1)* LENF); //第二维

printf("请输入约束条件系数:\n");

for (i=1;i<=M;i++)

    for (j=1;j<=N;j++)

        scanf("%f",&A[i][j]);
```

```

/*输入 b*/

b=(float*) malloc((M+1)*LENF);

printf("请输入对应顺序的 b:\n");

for (i=1;i<=M;i++) scanf("%f",&b[i]);


/*输入初始基可行解基变量所在的列号*/

BASIS=(int*) malloc((M+1)*LENINT);

STATUS=(int*) malloc((N+1)*LENINT);

for (i=0;i<=N;i++) STATUS[i]=0;//初始时所有的变量都是非基变量

printf("请输入初始基可行解基变量所在的列号:");

for (i=1;i<=M;i++)  {

    scanf("%d",&BASIS[i]);

    STATUS[BASIS[i]]=i;

}

SIGMA=(float*) malloc((N+1)*LENF);


}


/*两阶段法初始化*/

void initTwoPhase() {

    int i,j,n;//n 为初始时决策变量个数

    int kr;//剩余变量初始值

    int ka;//人工变量初始值

    printf("\n\n");

    printf("请输入线性规划问题初始相关信息:\n");

```



```
printf("请输入决策变量个数 n=");  
scanf("%d",&n);  
  
printf("请输入约束条件(已默认所有决策变量为非负)个数 M=");  
scanf("%d",&M);  
  
printf("请输入约束条件中\"<=\"类条件数:");  
scanf("%d",&DIRE[0]);  
printf("请输入约束条件中\">=\"类条件数:");  
scanf("%d",&DIRE[1]);  
printf("请输入约束条件中\"=\"类条件数:");  
scanf("%d",&DIRE[2]);  
  
N=n+DIRE[0]+DIRE[1];//原 LP 标准型下决策变量总个数  
NA=N+DIRE[1]+DIRE[2];//加入人工变量后决策变量总个数  
  
if (N==NA) {  
    printf("输入形式的线性规划问题未产生人工变量，请变形后再用两阶段  
法求解.");  
    exit (0);  
}  
  
/*此处不再要求初始输入为标准型，我们通过输入标准型，但通过程序将其  
化为标准型*/  
printf("请输入目标函数类型(1 for max、0 for min):");
```

```

scanf("%d",&objType);

C=(float*) malloc((N+1)*LENF);

printf("请输入目标函数价值系数:\n");

for (i=1;i<=n;i++) {
    scanf("%f",&C[i]);
    if (!objType) C[i]=-C[i];//当目标函数求 min 时，则转化为求 max
}

for (i=n+1;i<=N;i++) C[i]=0;


A=(float**)malloc((M+1)*sizeof(float*)); //第一维
for(i=1;i<=M; i++)
    A[i]=(float*)malloc((NA+1)* LENF);//第二维
BASIS=(int*) malloc((M+1)*LENINT);
STATUS=(int*) malloc((NA+1)*LENINT);
for (i=0;i<=NA;i++) STATUS[i]=0;//初始时所有的变量都是非基变量


/* 开辟第一阶段的价值系数矩阵 C1,并化成 max 型 */
C1=(float*) malloc((NA+1)*LENF);
for (i=1;i<=NA;i++) C1[i]=0;


kr=n+1;//剩余变量下标初始值
ka=N+1;//人工变量下标初始值


for (i=1;i<=DIRE[0];i++){

```

```

printf("请输入第%d 个\"<=\"条件的相关系数:",i);

for (j=1;j<=n;j++) scanf("%f",&A[i][j]);

for (;j<=NA;j++) A[i][j]=0;

A[i][kr]=1;//剩余变量

STATUS[kr]=i;//该剩余变量加入基

BASIS[i]=kr;

kr++;

}

for (;i<=DIRE[0]+DIRE[1];i++){

printf("请输入第%d 个\">=\"条件的相关系数:",i-DIRE[0]);

for (j=1;j<=n;j++) scanf("%f",&A[i][j]);

for (;j<=NA;j++) A[i][j]=0;

A[i][kr]=-1;//剩余变量

A[i][ka]=1;//人工变量

C1[ka]=-1;

STATUS[ka]=i;//人工变量做基变量

BASIS[i]=ka;

kr++;

ka++;

}

for (;i<=M;i++){

printf("请输入第%d 个\"=\"条件的相关系数:",i-DIRE[0]-DIRE[1]);

for (j=1;j<=n;j++) scanf("%f",&A[i][j]);

for (;j<=NA;j++) A[i][j]=0;

A[i][ka]=1;//人工变量

```

```
C1[ka]=-1;

STATUS[ka]=i;//人工变量做基变量

BASIS[i]=ka;

ka++;

}

/*输入 b*/

b=(float*) malloc((M+1)*LENF);

printf("请输入对应顺序的 b:\n");

for (i=1;i<=M;i++) scanf("%f",&b[i]);

SIGMA=(float*) malloc((NA+1)*LENF);

}

/*输出标准型*/

void output(float *C,int n) {

    int i,j;

    printf("Max Z=%0.1fX%d",C[1],1);

    for (i=2;i<=n;i++)

        printf("+0.1fX%d",C[i],i);

    printf("\n");
```

```
for (i=1;i<=M;i++) {  
    printf("%0.1fX%d",A[i][1],1);  
    for (j=2;j<=n;j++)  
        printf("%0.1fX%d",A[i][j],j);  
    printf("=%0.1f\n",b[i]);  
  
}  
for (j=1;j<=n-1;j++) printf("X%d,",j);  
printf("X%d>=0\n",n);  
}
```

/* 计算目标函数值 CN*b */

```
float computeObj(float *C,int n) {  
    float z=0;  
    int i;  
  
    for (i=1;i<=M;i++)  
        z+=C[BASIS[i]]*b[i];  
    return z;  
}
```

/*输出解*/

```
void displaySolution(float *C,int n) {
```

```
int i;

for (i=1;i<=n;i++){

    printf("X%d=",i);

    if (STATUS[i]==0)

        printf("0\n");

    else

        printf("%0.1f\n",b[STATUS[i]]);

}

printf("Max z=%0.1f\n",computeObj(C,n));

}
```

/*计算所有 σ */

```
void computeSigma(float *C,int n) {

    int i,j;

    float sum;

    for (i=1;i<=n;i++)

        if (STATUS[i]>0) {

            SIGMA[i]=0;

            continue;

        }

    else {

        sum=0;
```

```

        for (j=1;j<=M;j++)
            sum+=C[BASIS[j]]*A[j][i];
        SIGMA[i]=C[i]-sum;
    }

}

/*基变量中是否有非零的人工变量*/
int nonZeroArtiVarInBasis(int m,int n) {
    int i;
    for (i=1;i<=M;i++)
        if (BASIS[i]>m && fabs(b[i])>=1e-7) return 1;
    return 0;
}

```

/*判断是否存在某非基变量检验数为 0; 存在:返回 1 不存在:返回 0*/

```

int existNonbasisEqs0(int n) {
    int i;
    for (i=1;i<=n;i++)
        if (STATUS[i]==0 && fabs(SIGMA[i])<1e-7)

```

```
        return 1;

    return 0;
}
```

/* 单纯形法 C:目标函数价值系数; m:非人工变量个数 n:决策变量总个数*/

```
int simplex(float *C,int m,int n) {
    int i,j,flag,k,l;
    float theta,t;
    float pivot;//主元素

    while (1) {
        computeSigma(C,n);

        for (i=1;i<=n;i++) {
            if (SIGMA[i]>=1e-7) break; //存在 $\sigma_i > 0$ 
        }

        k=i;
```



```
/*所有 $\sigma \leq 0$ */  
  
if (i==(n+1))  
    /*无可行解，结束程序*/  
    if (nonZeroArtiVarInBasis(m,n)) {  
        printf("无可行解\n");  
        return 0;  
    }  
    /*无穷多最优解，结束程序*/  
    else if (existNonbasisEqs0(n)){  
        printf("无穷多最优解,其中一组解如下:\n");  
        displaySolution(C,n);  
        return 1;  
    }*/  
    /*唯一最优解，结束程序*/  
    else {  
        printf("有最优解:\n");  
        displaySolution(C,n);  
        return 1;  
    }  
  
  
/*检验是否有无界解*/  
  
flag=0;
```

```
for (;i<=n;i++) {  
    if (SIGMA[i]>=1e-7) {  
        for (j=1;j<=M;j++)  
            if (A[j][i]>=1e-7) break;  
        if (j==M+1) {  
            flag=1;  
  
        }  
    }  
    if (flag==1) {  
        printf("有无界解\n");  
        return 0;  
    }  
}
```

```
/*计算 $\theta$ */  
l=0;  
theta=MAX;  
for (i=1;i<=M;i++)  
    if (A[i][k]>=1e-7) {  
        t=b[i]/A[i][k];  
        if (t<theta) {  
            theta=t;  
        }  
    }
```

```

        l=i;

    }

}

/*  xk 为换入变量,l 行对应的基变量为换出变量  */
STATUS[BASIS[l]]=0;
STATUS[k]=l;
BASIS[l]=k;

/*旋转运算*/

pivot=A[l][k];
b[l]=b[l]/pivot;
for (j=1;j<=n;j++) A[l][j]=A[l][j]/pivot;//第 l 行操作
for (i=1;i<=M;i++)
    if (i!=l) {
        float pivot2=A[i][k];
        for (j=1;j<=n;j++) A[i][j]=-A[l][j]*pivot2+A[i][j];
        b[i]=-b[l]*pivot2+b[i];
    }
}

}

```

```
void release() {  
    free(A);  
    free(C);  
    free(b);  
    free(BASIS);  
    free(STATUS);  
    free(SIGMA);  
}  
  
void main() {  
    int choice;  
    int firstPhase;  
    //freopen("Two-phase.in", "r", stdin);  
    //freopen("2.in", "r", stdin);  
    printf("1.单纯形法\n2.两阶段法\n");  
    printf("请选择解 LP 的方法:");  
    scanf("%d",&choice);  
  
    if (choice==1) {  
        initSimplex();  
        printf("\n 标准型如下:\n");  
        output(C,N);  
        simplex(C,N,N);  
    }  
    else if (choice==2){
```

```

initTwoPhase();

printf("\n 初始标准型如下:\n");

output(C,N);

printf("加入人工变量后,第一阶段标准型如下:\n");

output(C1,NA);

printf("\n 第一阶段求解:\n");

firstPhase=simplex(C1,N,NA);

if (firstPhase) {

    printf("\n 第二阶段求解:\n");

    simplex(C,N,N);

}

else printf("即原 LP 无可行解\n");

}

//fclose(stdin);

release(); //代码中为了节省空间,使用了几个动态数组,按照规范,动态数组使用完毕需要释放

}

```

2. 运行窗口

(1). 例 2-6 用单纯形法求解线性规划问题

$$\begin{aligned}
 \max z &= 2x_1 + 3x_2 + 0x_3 + 0x_4 + 0x_5 \\
 \begin{cases} x_1 + 2x_2 + x_3 & = 8 \\ 4x_1 & + x_4 = 16 \\ 4x_2 & + x_5 = 12 \\ x_j \geq 0, j = 1, 2, \dots, 5 \end{cases}
 \end{aligned}$$

```

G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
1.单纯形法
2.两阶段法
请选择解LP的方法:1

请输入线性规划问题相关信息<已化为标准型,且已知初始基可行解基变量所在的列号>:
请输入决策变量个数N=5
请输入约束条件<已默认所有决策变量为非负>个数M=3
请输入目标函数价值系数:
2 3 0 0 0
请输入约束条件系数:
1 2 3 0 0
4 0 0 1 0
0 4 0 0 1
请输入对应顺序的b:
8
16
12
请输入初始基可行解基变量所在的列号:3 4 5

标准型如下:
Max Z=2.0X1+3.0X2+0.0X3+0.0X4+0.0X5
1.0X1+2.0X2+3.0X3+0.0X4+0.0X5=8.0
4.0X1+0.0X2+0.0X3+1.0X4+0.0X5=16.0
0.0X1+4.0X2+0.0X3+0.0X4+1.0X5=12.0
X1,X2,X3,X4,X5>=0
有最优解:
X1=4.0
X2=2.0
X3=0
X4=0
X5=4.0
Max z=14.0
Press any key to continue

```

(2) 例 2-9 线性规划问题

$$\min z = -3x_1 + x_2 + x_3$$

$$\begin{cases} x_1 - 2x_2 + x_3 \leq 11 \\ -4x_1 + x_2 + 2x_3 \geq 3 \\ -2x_1 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

试用两阶段法求解:

```

"G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
1.单纯形法
2.两阶段法
请选择解LP的方法:2

请输入线性规划问题初始相关信息:
请输入决策变量个数n=3
请输入约束条件<已默认所有决策变量为非负>个数M=3
请输入约束条件中"<="类条件数:1
请输入约束条件中">="类条件数:1
请输入约束条件中"="类条件数:1
请输入目标函数类型<1 for max、0 for min>:0
请输入目标函数价值系数:
-3 1 1
请输入第1个"<="条件的相关系数:1 -2 1
请输入第1个">="条件的相关系数:-4 1 2
请输入第1个"="条件的相关系数:-2 0 1
请输入对应顺序的b:
11
3
1

```

```

"G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
初始标准型如下:
Max Z=3.0X1-1.0X2-1.0X3+0.0X4+0.0X5
1.0X1-2.0X2+1.0X3+1.0X4+0.0X5=11.0
-4.0X1+1.0X2+2.0X3+0.0X4-1.0X5=3.0
-2.0X1+0.0X2+1.0X3+0.0X4+0.0X5=1.0
X1,X2,X3,X4,X5>=0
加入人工变量后,第一阶段标准型如下:
Max Z=0.0X1+0.0X2+0.0X3+0.0X4+0.0X5-1.0X6-1.0X7
1.0X1-2.0X2+1.0X3+1.0X4+0.0X5+0.0X6+0.0X7=11.0
-4.0X1+1.0X2+2.0X3+0.0X4-1.0X5+1.0X6+0.0X7=3.0
-2.0X1+0.0X2+1.0X3+0.0X4+0.0X5+0.0X6+1.0X7=1.0
X1,X2,X3,X4,X5,X6,X7>=0

```

```

"G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
第一阶段求解:
有最优解:
X1=0
X2=1.0
X3=1.0
X4=12.0
X5=0
X6=0
X7=0
Max z=0.0

第二阶段求解:
有最优解:
X1=4.0
X2=1.0
X3=9.0
X4=0
X5=0
Max z=2.0
Press any key to continue_

```

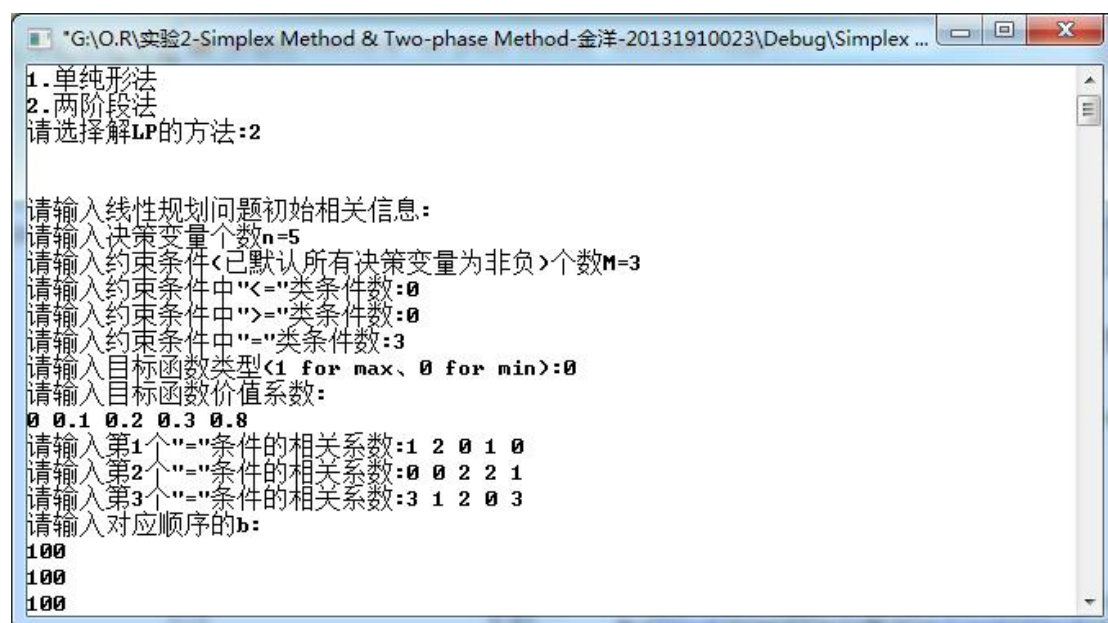
即原问题 $\min z = -2$

(3). 例 2-10 (合理利用线材问题)

$$\begin{aligned} \min z &= 0x_1 + 0.1x_2 + 0.2x_3 + 0.3x_4 + 0.8x_5 \\ \begin{cases} x_1 + 2x_2 + x_4 &= 100 \\ 2x_3 + 2x_4 + x_5 &= 100 \\ 3x_1 + x_2 + 2x_3 + 3x_5 &= 100 \\ x_1, x_2, x_3, x_4, x_5 \geq 0, \text{且为整数} \end{cases} \end{aligned}$$

试用两阶段法求解，当目标函数改为等价的 $\omega = x_1 + x_2 + x_3 + x_4 + x_5$ 时,解是否一致；

①




```

"G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
初始标准型如下:
Max Z=0.0X1-0.1X2-0.2X3-0.3X4-0.8X5
1.0X1+2.0X2+0.0X3+1.0X4+0.0X5=100.0
0.0X1+0.0X2+2.0X3+2.0X4+1.0X5=100.0
3.0X1+1.0X2+2.0X3+0.0X4+3.0X5=100.0
X1,X2,X3,X4,X5>=0
加入人工变量后,第一阶段标准型如下:
Max Z=0.0X1+0.0X2+0.0X3+0.0X4+0.0X5-1.0X6-1.0X7-1.0X8
1.0X1+2.0X2+0.0X3+1.0X4+0.0X5+1.0X6+0.0X7+0.0X8=100.0
0.0X1+0.0X2+2.0X3+2.0X4+1.0X5+0.0X6+1.0X7+0.0X8=100.0
3.0X1+1.0X2+2.0X3+0.0X4+3.0X5+0.0X6+0.0X7+1.0X8=100.0
X1,X2,X3,X4,X5,X6,X7,X8>=0

第一阶段求解:
有最优解:
X1=0
X2=40.0
X3=30.0
X4=20.0
X5=0
X6=0
X7=0
X8=0
Max z=0.0

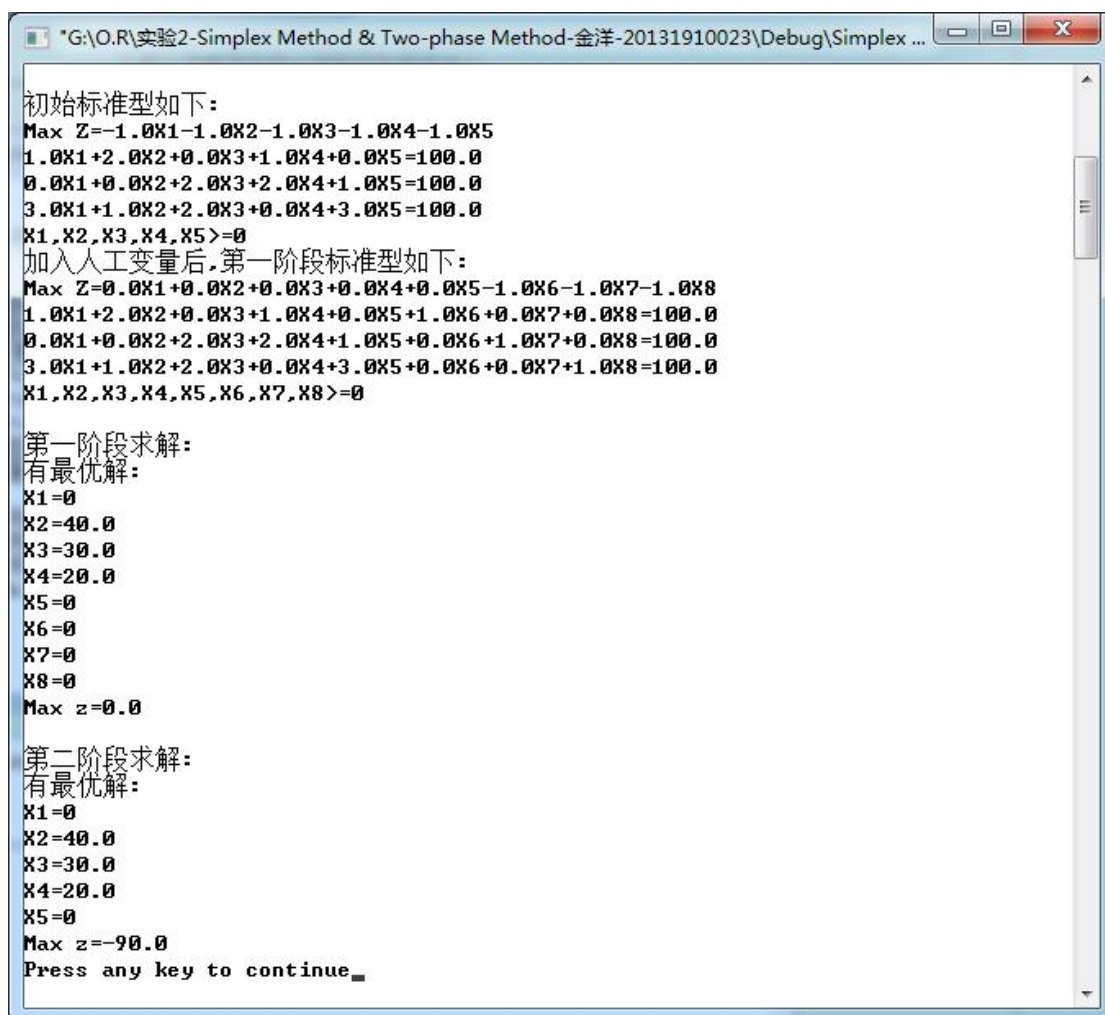
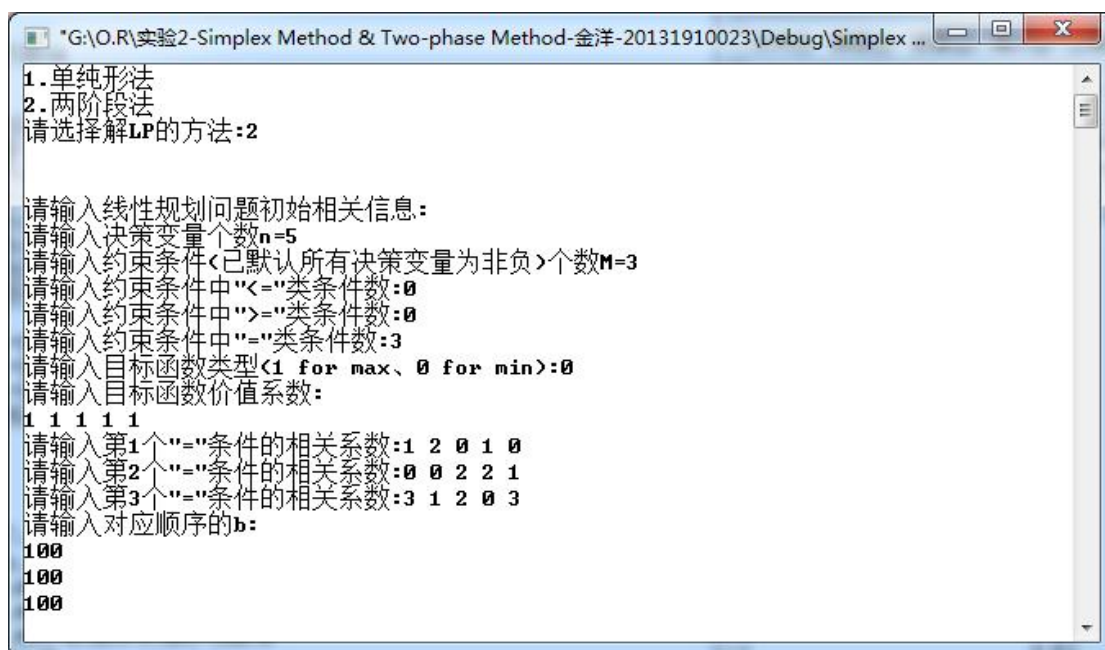
第二阶段求解:
有最优解:
X1=0
X2=40.0
X3=30.0
X4=20.0
X5=0
Max z=-16.0
Press any key to continue

```

此处得原 LP 的一个最优解 $x_1 = 0, x_2 = 40, x_3 = 30, x_4 = 20, x_5 = 0, \max z'' = -16$.

即原问题 $\min z = 16$. 这与课本上的解($x_1 = 30, x_2 = 10, x_3 = 0, x_4 = 50, x_5 = 0$.)有所不同, 但是把该解代入目标函数, $z = 0 \cdot 30 + 0.1 \cdot 10 + 0.2 \cdot 0 + 0.3 \cdot 50 + 0.8 \cdot 0 = 16$ 与程序解得的目标函数值相等, 故这两个都为最优解, 若没有整数限制, 由这两个解做凸组合, 将得到无穷多个最优解.

② 目标函数由 $\min z = 0x_1 + 0.1x_2 + 0.2x_3 + 0.3x_4 + 0.8x_5$ 换为 $\min \omega = x_1 + x_2 + x_3 + x_4 + x_5$ 时, 由于 $\frac{z + 100(2.9 + 2.1 + 1.5)}{7.4} = \omega$ 故两个目标函数其实是等价的, 在程序中求解, 如下:



可以看到, 求解结果与①中一致.

4. 作业题 2.6 (2), 用两阶段法求解下述 LP 问题:

$$\min z = 2x_1 + 3x_2 + x_3$$

$$\begin{cases} x_1 + 4x_2 + 2x_3 \geq 8 \\ 3x_1 + 2x_2 \geq 6 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

```

G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
1.单纯形法
2.两阶段法
请选择解LP的方法:2

请输入线性规划问题初始相关信息:
请输入决策变量个数n=3
请输入约束条件<已默认所有决策变量为非负>个数M=2
请输入约束条件中"<="类条件数:0
请输入约束条件中">="类条件数:2
请输入约束条件中"="类条件数:0
请输入目标函数类型<1 for max、0 for min>:0
请输入目标函数价值系数:
2 3 1
请输入第1个">="条件的相关系数:1 4 2
请输入第2个">="条件的相关系数:3 2 0
请输入对应顺序的b:
8
6

初始标准型如下:
Max Z=-2.0X1-3.0X2-1.0X3+0.0X4+0.0X5
1.0X1+4.0X2+2.0X3-1.0X4+0.0X5=8.0
3.0X1+2.0X2+0.0X3+0.0X4-1.0X5=6.0
X1,X2,X3,X4,X5>=0
加入人工变量后,第一阶段标准型如下:
Max Z=0.0X1+0.0X2+0.0X3+0.0X4+0.0X5-1.0X6-1.0X7
1.0X1+4.0X2+2.0X3-1.0X4+0.0X5+1.0X6+0.0X7=8.0
3.0X1+2.0X2+0.0X3+0.0X4-1.0X5+0.0X6+1.0X7=6.0
X1,X2,X3,X4,X5,X6,X7>=0

第一阶段求解:
有最优解:
X1=0.8
X2=1.8
X3=0
X4=0
X5=0
X6=0
X7=0
Max z=0.0

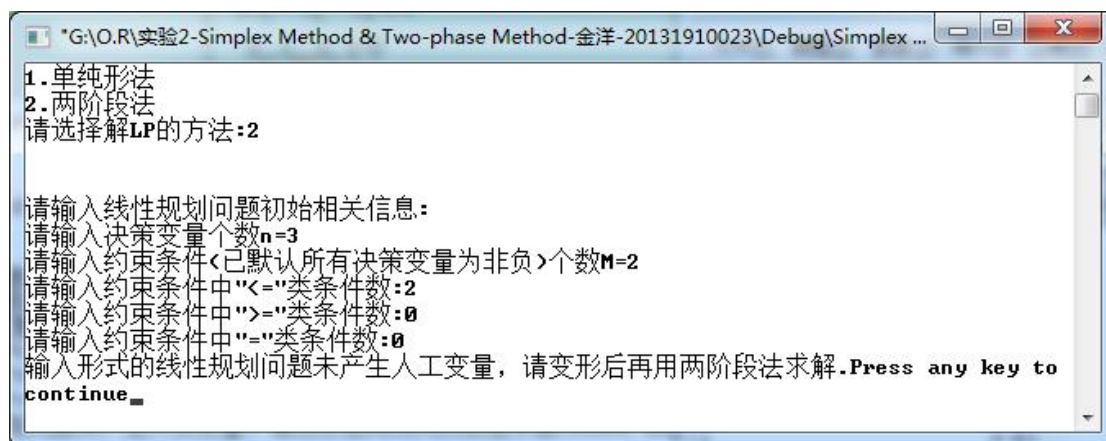
第二阶段求解:
有最优解:
X1=0.8
X2=1.8
X3=0
X4=0
X5=0
Max z=-7.0
Press any key to continue

```

即原问题 $\min z=7$.

5. 例 3-4 求解线性规划问题

$$\begin{aligned} \max z &= x_1 + x_2 \\ \begin{cases} -x_1 + x_2 + x_3 \leq 2 \\ -2x_1 + x_2 - x_3 \leq 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$



经过观察分析，两个约束条件加入剩余变量后即产生了一组基，不必再次加入人工变量，没有必要使用两阶段法。我们将其变形，两个约束条件改变不等号方向，可以使用程序进行两阶段法求解：

```

G:\O.R\实验2-Simplex Method & Two-phase Method-金洋-20131910023\Debug\Simplex ...
1.单纯形法
2.两阶段法
请选择解LP的方法:2

请输入线性规划问题初始相关信息:
请输入决策变量个数n=3
请输入约束条件<已默认所有决策变量为非负>个数M=2
请输入约束条件中"<="类条件数:0
请输入约束条件中">="类条件数:2
请输入约束条件中"="类条件数:0
请输入目标函数类型(1 for max、0 for min):1
请输入目标函数价值系数:
1 1 0
请输入第1个">="条件的相关系数:1 -1 -1
请输入第2个">="条件的相关系数:2 -1 1
请输入对应顺序的b:
-2
-1

初始标准型如下:
Max Z=1.0X1+1.0X2+0.0X3+0.0X4+0.0X5
1.0X1-1.0X2-1.0X3-1.0X4+0.0X5=-2.0
2.0X1-1.0X2+1.0X3+0.0X4-1.0X5=-1.0
X1,X2,X3,X4,X5>=0
加入人工变量后,第一阶段标准型如下:
Max Z=0.0X1+0.0X2+0.0X3+0.0X4+0.0X5-1.0X6-1.0X7
1.0X1-1.0X2-1.0X3-1.0X4+0.0X5+1.0X6+0.0X7=-2.0
2.0X1-1.0X2+1.0X3+0.0X4-1.0X5+0.0X6+1.0X7=-1.0
X1,X2,X3,X4,X5,X6,X7>=0

第一阶段求解:
有最优解:
X1=1.0
X2=3.0
X3=0
X4=0
X5=0
X6=0
X7=0
Max z=0.0

第二阶段求解:
有无界解
Press any key to continue_

```

第二阶段求解为无界解，这与观察一致，如 $(+\infty, 0, 0)'$ 便是原 LP 的无界解。

六、总结

1. 学会使用单纯形法对线性规划问题求解；
2. 学会使用 c 语言实现单纯形法；
3. 在原问题没有直接给出初始基可行解时，添加人工变量后，利用两阶段法可以进行求解；
4. 学会使用 c 语言实现两阶段法；

七、参考文献

- [1] 谭浩强著,《c 程序设计》(第三版),清华大学出版社,2005.7;
- [2] 《运筹学》教程编写组,《运筹学》(第4版),清华大学出版社,2013.1;
- [3] 吴振奎,钱智华,于亚秀著,《运筹学概论》,哈尔滨工业大学出版社,2015.2;

八、教师评语